

COMAL Standardization and Development Group Meetings

Velje, Denmark. 14th to 16th August 1986.

1 Attendance

Bason	Frank	(fb)	UniComal
Burkinshaw	Peter	(pb)	TeleNova
Christensen	Borge	(bc)	
Clack	John	(jc)	PNL (Secretary)
Grainger	Brian	(bg)	
Havn	Ib	(ih)	Dansk System Elektronik a/s
Hoedholt	Herman	(hh)	
Hojsholt-Poulsen	Leo	(lh)	
Hyland	Bill	(bh)	Irish Department of Education
Jensen	Jens Erik	(jej)	UniComal a/s
Kjaer	Mogens	(mk)	
Kristiansen	Freddy	(fk)	DalgaSoft
Lassin	Helge	(hl)	UniComal a/s
Laurson	Lars	(ll)	
Nielson	Kirsten	(kn)	Regnecentralen
Pirie	David	(dp)	Glasgow University
Purbe	Sven	(sp)	TeleNova
Quin	Angus	(aq)	Paisley College of Technology
Ronhoj	Erik	(er)	Dansk System Elektronik a/s
Schwartzlose	Johnny	(js)	
Tjur	Stig	(st)	Dansk Data Elektronik

## 2 Welcome To The Horstruup Centre, Velje

jej welcomed those present to the Hornstruup Centre, Velje, Denmark.

## 3 Administrative Arrangements

### 3.1 Election Of Chair

There being no chair, nominations were invited. Angus Quin was duly elected to serve as chair for this session

### 3.2 Secretary's Apologies

jc apologised for the lack of communication, offering problems with the handover from the previous secretary. He undertook to stand again if this was acceptable. This being the case, jc continued as secretary.

### 3.3 Agenda

Since there had not been an agenda circulated, aq from the chair suggested an Agenda, with priorities. This was broadly agreed.

### 3.4 Previous Meeting's Minutes

The minutes of the previous meeting had been previously been circulated. The meeting AGREED to adopy them as a true record (with minor typographical amandments), and they were duly signed.

### 3.5 Administrative Costs

The Secretary reported he had not been able to gain access to the Comal funds, and was so not in a position to recoup his expenses! He was encouraged to pursue this matter.

### 3.6 Changes In Group Membership

fk lapsed his membership of the Standardization Group and was attending as an observer.



#### 4 State Of COMAL Today

aq said that he felt COMAL was now growing up, and had applications in the commercial and industrial world. bh said that he felt it had strong appeal in its ability to facilitate language learning. There was a need for a pedagogical approach; the language's extensions should be made in such a way as to make it applicable for a wide range of uses.

kn saw COMAL as an educational language. Her interest was to strengthen the educational ideas for school and college.

st said that his company sold into business schools, and so saw COMAL as a mainly educational language, but recognized that some college graduates were now requiring COMAL for business.

dp said that although currently using UniComal Commodore 64 COMAL for teaching, he would like to see the language extended to data acquisition and process control.

bg would like more technical support for disc-based COMAL and UniComal

ih said they were writing industrial systems in COMAL, and were using i/o systems for process control written in COMAL.

er supported the use of COMAL in industrial applications.

hl said that UniComal (Commodore and IBM) sales were into the educational and industrial market

sp said most (90%) of TeleNova's sales were to schools and colleges in Denmark and Sweden, and that COMAL was 'bundled' with the hardware.

fk said that he implemented COMAL on 280 CPM systems, and was working on a cartridge for the Amstrad 8xxx series.

lh said that he was working on COMAL documentation for the UniComal C64 cartridge.

bh said he worked with the Ministry of Education in Dublin.

aq said that he was a lecturer, and worked with a standardization group on graphics.

pn said he was an COMAL implementer

#### 5 Publicity

It was noted that there had been a lack of publicity. It was felt that COMAL had been overshadowed by BBC Basic in the UK, and that IBM (UK) had not appreciated the demand for it. dp said that as far as he knew Glasgow was the only university to be using COMAL as a teaching language. He understood that Scotland has chosen COMAL as one of their teaching languages for Schools Certificate. lh said that more articles on COMAL were needed. The use of free discs to distribute the language was felt by aq to be a very good way of getting the language moving, and to introduce it to commercial and industrial users. Roy Thornton and Ian Malcom were understood by pb to be giving a talk to MUSE (an English Micro's in Education Group)

##### 5.1 Product News



5.1.1 UniComal - jej reported there was a new version of the Commodore C128 cartridge, which would use the new features of the Commodore 128. They were developing COMAL for the 8086/88 series of microprocessors. had been some lobbying on Logo's behalf.

5.1.2 RegneCentralen (RC) - kn said that a new version of COMAL came out in Spring 86. Inspired by UniComal's packages, a new version with packages was aimed for release in Fall 86.

5.1.3 DDE - st said that they now had a version of COMAL running under Unix v5. They were working on packages, and hoped to have this feature available at some time.

5.1.4 TeleNova - pb said that they were releasing COMAL 2.1 (which includes most extensions). They had also implemented a serial port product. Packages were coming along, as was the user documentation. A 8087 version of COMAL could be expected soon. It was hoped to implement dynamic records and store management.

5.1.5 Glasgow University - dp reported that the Aeronautic Department had moved from Basic to COMAL on the Commodore 64's. They were hoping for some COMAL implementations for the IBM and Apricot pc's and Vax. COMAL is regarded as a teaching language and a professional language.

5.1.6 Portsmouth Grammar School - rt reported on a micro users conference he had attended in Nottingham. He thought Logo users had closed minds, and in their enthusiasm to get away from Basic were misguidedly treating COMAL as another Basic.

bc commented that Logo is a refuge from Basic, and is not all bad. He reported on a visit he made to Jet Propulsion Laboratory (JPL), Pasadena, where he talked about COMAL. There had been demonstrations on a Commodore 64 and an IBM pc. bc said he was impressed by the speed of UniComal.

5.1.7 DalgaSoft - fk reported that he had completed a cartridge for the Amstrad 8xxx, incorporating packages, and error messages in three (3) languages. It also featured graphics, sound, and external procedures. He was awaiting hardware so that he could start work on a new CPM version that would work across the range of CPM machines. A disc version was also available.

5.1.8 Dansk System Elektronik - er said that they had written an interface to a communications board in COMAL and an i/o system, using packages.

5.1.9 General -

5.2 Marketing And Distribution

fk said that he was looking for an UK distributor.

5.3 Published Standard

bh pointed out that the absence of a published standard made it difficult for people to know what the language comprised. dp said that the textual representation in the TCD kernal was not as clear as a bnf diagram

6 Agenda

At this point discussion took place on an Agenda for the next two days. The following topics were agreed; the order reflecting the priority placed on them.

1. Packages
2. TeleNova Proposals
3. Graphics
4. Data Structures
5. Extensions - facilities vs implementation
6. Real time and Concurrent COMAL
7. The Standard

The IN operator, it was pointed out, had been considered by the Standardization Group at the Edinburgh meeting, and so it would not be worthwhile considering it again.

7 Packages

er said that the point of packages should be to hide complexity from the user, who is not concerned with the hardware. He gave as example a function to return voltage from a circuit, eg VoltageIn (Card, Channel). The native language of the package should not matter (it may be machine code) but one would like to be able to write packages in COMAL. The use of packages does raise the need for local static variables. He suggested that packages should be linked to the user's COMAL program or the environment. bg saw the need for packages to meet specific needs, eg device drivers. He thought they should be (able to be) written in machine code for speed.



Implementors may wish to use packages to hide away code from the user to prevent unwanted alterations, and the user would get only the wanted functionality (at machine code speed).

jej said that packages could be used to extend the language, specific to an application. A teacher could create packages as a series of building blocks for the students, eg sorting routines.

At this point bg said that the difference between procedure and packages was not clear. kn said that the difference was that packages brought much more into the user environment. hl raised again the importance of closed static variables in packages, pointing out that otherwise the variables must be made global.

bg raised the question of speed again, and wanted to know if packages ran at 'COMAL' speed. jej said that they did on the IBM pc. bh pointed out the difference of emphasis on packages between manufacturers, who wanted to hide the code, and for users, who wanted to make public domain software easily available. er pointed out that packages allowed programmers to choose their own level of abstraction, and ib pointed out that they could make things easier for a beginner.

bh again stressed the difference between commercial and public need - the closed box vs open code. er pointed out that the source code is often made available with the package - the goal is not to hide it away.

At this point kf asked how packages were to be implemented; pb said that it would take too long to explain. dp said that he felt the important thing was not how packages were implemented, but that they should be portable across machines. js said that the group should be concerned how users are allowed to implement their own packages. However, fk felt that package implementors were 'experts', with access to machine-specific manuals. kn pointed out the difference between COMAL and non-COMAL packages.

mk felt that there were a number of basic issues which should be considered.

1. how packages are got into the machine
2. how packages are initialized within the program
3. how packages are accessed from the program
4. how packages are disposed of
5. error handling with packages
6. how packages are used

It was agreed to this as a structure for discussion.

Further points to be considered were:

7. what tools were to be available for package writers
8. what enquiry functions should there be

jej said that there should be an enquiry function within a package to allow one to list (say) procedure names. bh wondered to whom these features were addressed.

kn said that their enquiry functions were:

ShowPackage - shows packages used

ListPack Package - show procedures and functions used in a package, with the heading (ie, parameter list)

It was AGREED that this was a useful feature to provide as standard for users.



-----  
The Development Group broke for lunch and resumed Thursday afternoon  
-----

## 8 Proposals For Changes To The Extension

### 8.1 TeleNova Proposals - 1985-05-30

It was agreed pb should speak on the proposals, and that there should be general discussion before any votes were taken. It was suggested that the desirability of the Proposal be considered, as well as the syntax proposed.

#### 8.1.1 Proposal 1 Page 1 - Array Alias -

pb said that it was to simplify the passing of an array sub-set, and could be considered to be similar to the Fortran "Equivalence".

kn said that she found it too complicated - it was only needed for passing parameters to procedures, and was too complicated for the ordinary user to understand the implementation. bg found the use of the DIM keyword confusing, as the variable 'a' appeared in two DIM statements. pb said that it was not practicable to have a separate DIM for 'b'. mk asked if 'b' could be re-aliased, and was told it could not. pbn was asked if sub-array assignment was the only use, and replied that it could also be used for initializing slices of arrays which were not necessarily contiguous (but this feature could not be allowed under the current standard). jc pointed out the Fortran Equivalence statement had been considered for deprecation on the grounds that it reduced portability. pb said that his emphasis was on problem solving, not portability. Implementations storing n-dimensional arrays in different ways will not be compatible. bh suggested another keyword, which bg supported. mk suggested DIM AT.

#### 8.1.2 Proposal 2 Page 2 - Array Assignment -

pb said that for orthogonality he would have preferred

```
20 a:=1
```

but would optionally allow

```
20 a():= 1
```

fk asked if this implied '( , , )' for 3-D arrays, and er asked if '( , )' would be used for the first array element, with '( , , )' for the others. ib spoke strongly for the use of '()', and pn said that he was happy for the use of brackets to be optional. mk asked if the use of parentheses for arrays in formal parameter lists was optional, and was assured by pb that it was.

st asked if the construction

```
b(*,2) := 5
```

could be considered to have the effect of assigning the value '5' to b(1,2), b(2,2), b(3,2) etc.

8.1.3 Proposal 3 Page 3 - Array Input/output -

kn thought that this was generally desirable, but foresaw syntactical problems, particularly with n-dimensional arrays. bg felt this feature would be most useful for file input and output. hl wanted to know what happened on printing when the array subscript changed, and bg wanted to know if again the parentheses were optional.

8.1.4 Proposal 4 Page 3 - Repeat Factor In USING Strings -

It was asked how the string "10 would be output and pb suggested the metacharacter "'10". There was then discussion about internal nesting, which is not allowed at present. ll suggested an internal nesting mechanism.

8.1.5 Proposal 5 Page 3 - Multiple Assignment -

This feature was agreed desirable, but nk did not like the idea of mixing scalars and arrays. mk felt the language should be kept clean and didn't like the multiple assignment construction. hl was concerned with the problem of upward compatibility for beginners. er suggested the construct

a:=b:=c:=3

but this is not feasible, as ':=' and '=' are interchangeable in Comal, and the above statement could be construed as

a:= {TRUTH OF (b=3)}

8.1.6 Proposal 6 Page 4 - Negative Substring Specifiers -

It was pointed out by mk that the substring specifier 0 was rejected at the last meeting - item 6.2.21 refers. pb suggested using 0 as a cipher for the position after the last character; mk thought it was better to introduce something new, while bg thought that the use of -1 as a substring specifier was a mistake!

8.1.7 Proposal 7 Page 4 - Print Separators -

The desirability of these was agreed. However, ip thought that it made programs more difficult to read, and it was better to use two PRINT statements. bg said this could lead to misuse. hl said that there was a need for a separator which did nothing, also for one which produced a line-feed. He suggested

, (comma) - does nothing

; (semicolon) - start in next zone

and that the default zone is 1.



8.1.8 Proposal 8 Page 5 - Compound Statements -

The desirability was in question. mk said that although their COMAL had this feature he did not like it. hl said that it should not be allowed for procedure calls and assignments. ip said that they would not support this, but ll said that the natural grouping of procedure calls made for readability. The problem of how error messages identified the offending component of a compound statement was raised by kn.

8.1.9 Proposals 9 & 10 Page 5 - MAXIMUM And MINIMUM -

pb said that the function could accept one argument if it was an array. er said that he felt this problem was best solved by using a package. ll felt that they should be provided as in-built functions. hh said that both functions were needed, and suggested two new functions - MAXROW and MINROW - which would return the address of the column or row containing the largest value. At this point pb said that he saw a need to clarify requirements. js said MAXIMUM and MINIMUM were the functions most needed, and so had the higher priority, but suggested that a three (3) letter abbreviation more consistent.

8.1.10 Proposal 11 Page 5 - SIGMA -

pb explained that the main purpose of this function was to allow the fast processing for a frequently required operation. He commented that Roy Thornton, in his note, wanted too much from a simple function. ll said that he saw this as a popular function.

8.1.11 Proposal 12 Page 6 - Second Parameter To STR\$ -

fk said that he would prefer a USING statement. er said that COMAL lacks an output facility; and although this was an aid it would be better in a PRINT USING statement. The use of trailing zeroes (0's) was explained by pb to be for improved readability.

8.1.12 Proposal 13 Page 6 - REF Operator -

mk felt that the choice of '#' (hash) for a pointer type was poor, as it could be confused with an INTEGER type; he thought '@' would be more appropriate. dp thought that this item would be better discussed in the context of data structures. pb thought that REF would be a useful precursor to packages which carried out certain operations. It was also a valuable acquisition for COMAL on the way to dynamic data structures.

8.1.13 Proposal 14 Page 7 - Subarrays -

jk felt that it would again be essential in the context of addressing package procedures. kn thought the syntax complicated, and that the proposal did not say how multi-dimensional arrays would be handled.

The following proposals are from the TeleNova Proposal Document (II)

8.1.14 Proposal 15 Page 1 - Hex Constants -

pb stated that he had changed the sign from a '&' (ampersand) to a '\$' (dollar) since there were already COMAL implementations with this feature. er asked if this feature could be provided to any base, but pb replied that that would be too difficult to implement. kn said that this feature would be better as a package, but until they arrived this could be a way to incorporate this facility. er agreed that any base would be harder, but pointed out the need for Octal (base 8) in his applications.

8.1.15 Proposal 16 Page 1 - Hex Strings -

pb said that these had a use when outputting printer codes.

8.1.16 Proposal 17 Page 1 - Bit Constants -

These were said to be useful for graphics, but a higher and lower limit would need to be specified by the implementors.

8.1.17 Proposal 18 Page 2 - Variable Length Integers -

It was felt to be very important for machine portability, but that the naive user should not be concerned with this feature; some sensible default being available.

The Proposals were then put to the vote. It was agreed to vote first on the DESIRABILITY of the feature given by the Proposal, and then on the implementation as given in the Proposal.

Proposal	Comments	Votes		Outcome
		Desirability	Syntax	
1		4-11		FAILED
2		15-1		AGREED
2.1	without ()		4-8	FAILED
2.2	optional ()		3-9	FAILED
2.3	mandatory (), opt ,		1-11	FAILED
2.4	mand (,,)		12-0	AGREED



COMAL Standardization and Development Group Meetings  
Proposals For Changes To The Extension

Page 1k

2.5	opt (,,)		0-12	FAILED
3		15-1	12-0	AGREED
4		7-16		FAILED
5		10-6		FAILED
6		2-14		FAILED
7	no vote taken - deferred for further discussion			
8		2-14		FAILED
9 & 10		10-4		FAILED

Proposal	Comments	Votes		Outcome
		Desirability	Syntax	
11		3-13		FAILED
12		2-14		FAILED
13	no vote taken - deferred for further discussion			
14		14-2	3-13	FAILED
15		15-1	15-1	CARRIED
16		14-2	14-2	CARRIED
17		16-0	16-0	CARRIED
18		7-11		FAILED

-----  
The development group then broke for dinner  
-----

The Development group resumed on Friday 15th

The secretary recalled the votes of the Development Committees.  
It was agreed to relate the UniComal proposals to the UniComal document 'A PROPOSAL FOR PACKAGES'.

jej said that the items referred to as A(i), A(ii) and A(iii) were covered in the proposal. A(iv) was covered by the DISCARD statement. Item A(v) could not be addresses without a discussion on error-handling. dp wished to know if one could selectively discard. jej replied that

DISCARD - in kernel

DISCARD <list> - for the extension

He also noted that PACKAGES made static binding essential.

aq said that he regarded item A had been covered in sufficient depth, and moved the group onto item B.

jej made a few extra points on this item - that one does not have to merge code into all external procedures. This would make updating difficult. If a package is used, then just the package need be updated, and all references to that package will automatically be updated. bgra observed that the UniComal proposal didn't cover item C. dp asked how one would be able to edit modules. jej replied that they would be treated as normal COMAL programs, and could be edited and saved. jej said that 'black-box' packages could be distributed if people so wished, but of course one could also distribute the source.

bh saw three distint situations -

the 'black-box' - no source - for the commercial environment

it was possible (but difficult) to edit the source

the source was not protected

and felt that there should be three separate ENVIRONMENT commands for these situations.

Item E (ii)



jej proposed that implementor should move from dynamic to static binding for all COMAL programs, and not just packages. mk stated that UniComal was prepared to make this move, especially with the need to allow COMAL to be used for concurrent processing. pb pointed out the impact that this could have, but the mood was that a move to static binding would affect a very small proportion of COMAL users. The point was made that static binding also gave the benefit of speed of execution. A straw poll indicated that:

Static binding	Dynamic binding
Mytech	RC
DDE	TeleNova
Metanic	AcornSoft
	DalgeSoft
	UniComal

bh said that he believed compiled versions of languages usually used static binding. dp pointed out that there would be problems of portibility across static and dynamic variable implementations of COMAL, but jc thought that users must accept this. jej made the point that if compiled COMAL was wanted then it must use static variables. hh thought that there should be a statement on binding in the kernel. A discussion followed, with kn making the point that the comiler/interpreter must do more work, and by neccessity be slower with static binding, with, as pb pointed out, a consequently larger program.

At this point bh asked for an indication of the implications in the comparison of static vs dynamic binding. kn said that manufacturers shoul indicate which binding is used, the implications, and clear guidance to COMAL program writers on how to ensure portability.

pb said that he would write a paper which would give general guidance on what to avoid doing when writing for portability.

Page 3

mk said that a USE statement would make everything visible in the block - and it should be mandatory that this was the first statement. The problems of mixed-language programming were discussed, in particular the problems of parameter passing, and mg felt it should be possible. However, pb saw practical difficulties, and it was agreed that not all languages could be interfaced with COMAL.

kn then proposed that discussion on - pages 2-4 - the scope rules - be deferred.

## 8.2 UniComal Proposal

kn said that the group must defer discussion of scope to the next meeting. er said that a DISCARD statement was needed urgently in the language.

## Semantics

mk proposed that USE should be the first statement in the block, but if the user does not comply COMAL should gather all USE statements and execute them AS IF they were at the beginning of the block. pb thought that the usefulness of USE was with scope, otherwise a LINK statement would seem more appropriate. ll said that USE would free the users from having to use a dot notation if multiple USE statements gave a symbol name clash. jej felt that USE changed the program environment, and that there was a need to detect if the environment has been initialized or not. hl felt that there should be provision to allow the user to CONTINUE processing. jej said that surely the COMAL environment must tell all packages what its current state is. bh felt a working party was needed, and jej agreed that the implementors should meet. The Danish implementors AGREED to hold meetings, and to attempt to involve and keep informed their Swedish counterparts.

## 8.3 Page 8

This working paper was welcomed as a constructive approach to packages, and it was suggested by jej that this form the basis for the working party.

Borge Christensen then gave his paper on concurrent COMAL.

## 9 Real-time COMAL

lp suggested the following primitives :

a need for semaphores (with counters)  
    SIGNAL           (semaphore)  
    AWAIT           (semaphore)

and for synchronicity

    SEND           (message)  
    RECEIVE       (message)  
and for establishing priorities for tasks and messages  
    PRIORITY

er said that all data was global - but TASKS needed to be closed. The only means of communication was through a signal or SEND. He said that there was a need for re-entrant COMAL code, with closed scope. ip pnted out that there needed to be a

    SAVE COMAL STATE  
and  
    RESTORE COMAL STATE



to facilitate interrupt handling.

pb felt that this could be very time consuming within COMAL; er said that he was not asking for an implementation like Concurrent Pascal, as that was far too slow to use, and was not the best way to go. bc then asked for his program to be rewritten using ip's suggested statements. er said that resource allocation was a very important issue. jej said that even Concurrent DOS does not have this, but er said that for their applications they would like to have say 5 programs active. pb asked who would design concurrent COMAL. ip identified two areas for designers:

the passing of messages,  
and  
the synchronization of processes.

bc asked for the matter to be given very careful consideration. mk said that he would not implement the suggested KEYWORDS into the COMAL standard without a concurrent operating system. er and jej (UniComal) AGREED to collaborate on a draft Concurrent COMAL, with bc to act as a consultant. The other Danish implementors AGREED to also be involved in this undertaking.

#### 10 UniComal Print Statements

The Development Group then sat

bc gave a historical overview.

Under DataGeneral X-Basic, the separators had the following effect -  
; made a space for digits, no space for characters  
, always moved to the next zone

So, in 1980, it was decided that ; would always give a space, and that , would NEVER give a space. Then, ZONE was introduced so that one could then have item separators, which would NOT have the effect of PRINT formatters. jej said that UniComal has used the PUTFILE statement because users could not rely on the ZONE statement giving the desired effect. pb preferred ,, to indicate 'goto a new ZONE', but sj felt that an entirely new separator was required. bc felt that a new separator was a good idea, and that the main obstacle were the historical reasons. dp suggested that where ZONE had a value greater than zero PRINT formatting should take place, but that a ZONE value of zero should be taken as a cipher for no action. hl pointed out that when ZONE is not equal to zero there would be no way of concatenating output. bc stressed that it was important not to take away the possibility of the comma meaning 'do nothing'.

#### 11 Date Of Next Meeting

Ian Malcolm had offered to arrange another meeting in Edinburgh, but at the request of the meeting bh undertook to investigate the possibility of a meeting in Ireland (Dublin). aq felt that the group should aim for a meeting next spring - say March/April - during the academic vacation. aq undertook to

investigate the possibilities of meetings in Holland/Dublin/London/Edinburgh.

#### 11.1 Finance

jc said that no money had been passed on to him. kn recollected that subscriptions had last been collected in 1984. jc undertook to attempt to recover the COMAL account, and to invoice companies for the 1986/87 year. The current membership charge of 250 pounds sterling was to be maintained.

The development group then resumed.

#### 12 TeleNova Proposal

bc asked what its use was - pb replied that users should be allowed to (easily) print out the contents of an entire array. er replied that PRINT USING should be enhanced. ip felt the proposal would make for less intelligible code, all to save one or two line of COMAL code. pb pointed out the PRINT () gave one element of the array on each line. bc made the point that vertical code was always to be preferred to horizontal code. The motion was DEFEATED by 5-12.

At this point the secretary left.

John Clack  
(Secretary)  
1 September 1986